INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Numerical solution of optimization test-cases by Genetic Algorithms

Nathalie Marco-Blaszka and Jean-Antoine Désidéri

N° 3622

Février 1999

THÈME 4

Rapport de recherche

ISSN 0249-6399

Numerical solution of optimization test-cases by Genetic Algorithms

Nathalie Marco-Blaszka* and Jean-Antoine Désidéri*

Thème 4 — Simulation et optimisation
de systèmes complexes
Projet Sinus

Rapport de recherche n° 3622 — Février 1999 — 23 pages

Abstract: In this report, we present the numerical solution of four optimization problems by Genetic Algorithms (GAs). The test-cases involve two single-objective and two multi-objective optimization problems. In all four cases, the analytical functions to be optimized present a large number of local optima and the GA is demonstrated to be the most adequate optimizer. These four test-cases are part of the database developed within the INGENET European thematic network.

Key-words: Optimization - Genetic Algorithms - Steepest descent method - Multi-Objective Optimization - Criterion of non Inferiority - Pareto front

* INRIA, 2004 Route des Lucioles, BP. 93, 06902 Sophia Antipolis Cédex-France

Unité de recherche INRIA Sophia Antipolis
2004, route des Lucioles, B.P. 93, 06902 Sophia Antipolis Cedex (France)
Téléphone : 04 92 38 77 77 - International : +33 4 92 38 77 77 — Fax : 04 92 38 77 65 - International : +33 4 92 38 77 65

Résolution numérique de cas-tests d'optimisation par algorithmes génétiques

Résumé : Dans ce rapport, nous présentons la résolution numérique de quatre problèmes d'optimisation au moyen d'algorithmes génétiques. Les cas-tests correspondent à deux problèmes d'optimisation à un critère et deux autres problèmes multicritère. Dans les quatre cas-tests, les fonctions analytiques que l'on optimise présentent un grand nombre d'optima locaux. Il s'avère que les Algorithmes Génétiques se révèlent les mieux adaptés à la résolution de tels problèmes d'optimisation. Ces quatre cas-tests font partie de la base de données développée dans le cadre du réseau thématique européen INGENET.

Mots-clés : Optimisation - Algorithme génétique - Méthode de plus forte descente - Optimisation multicritère - Critère de non-dominance - Front de Pareto

Contents

1 Introduction

When an analytical function has to be optimized, many criteria can be taken into account to select the adequate optimizer : is the optimum local or global ? How many design parameters should be introduced ? What should their type(s) be ? Are there constraints or not ? Is the function convex or not ? Is the function differentiable ? Is it a multi-objective optimization ?

There exist a large number of optimizers : deterministic ones as gradient-based methods, Newton methods, quasi-Newton with interior points methods, one-shot methods or stochastic ones as Evolution Strategies (ES)[1] or Genetic Algorithms (GAs).

When a function exhibits a large number of local optima, unless the initial guess in the deterministic method is very close to the global optimum, the solution may converge to a local optimum. GAs (and ES) are potentially more robust to avoid this kind of problem because they operate on a population of individuals (and not a unique one as for deterministic methods). The population first explores the whole admissible search space and second, after a consequent number of generations, exploits the best individuals to converge to the global solution.

In this report, the basic GAs are assumed to be known and many references to the literature are provided (one can refer to D. Goldberg's book [4] or to one of our INRIA report [7]). We present four optimization problems involving analytical functions : two single objective and two multi-objective problems. The functions present a large number of local optima and they are very difficult to be optimized by deterministic methods. These four optimization problems are four test-cases under investigation within the INGENET European thematic network[2]. The first single optimization problem is solved with a Hybrid GA (GA coupled with a gradient-based method), the second one is solved by another Hybrid GA (GA coupled with a hill-climbing method) and the two multi-objective problems are also solved by GAs, by construction of Pareto fronts.

[1]ES appeared in the 1960s at the Technical University of Berlin, developed by Bienert, Rechenberg and Schwefel

[2]INGENET : *NET*worked *IN*dustrial Design and Control Applications using *GE*netic Algorithms and Evolution Strategies (Industrial and Material Technologies Programme, Brite EuRam III)

2 Rastrigin function

2.1 The minimization problem

The goal is to minimize the highly multimodal analytical function, called Rastrigin function [10]:

$$F(x) = n + \sum_{i=1}^{n}(x_i^2 - cos(2\pi x_i)) \quad \text{with } x = (x_i)_{i=1\ldots n} \quad \in \quad [-5.12, 5.12].$$

Figure 1 depicts the Rastrigin function in the case where $n = 2$. This function exhibits a large number of local minima.

Rastrigin function in the 2D case

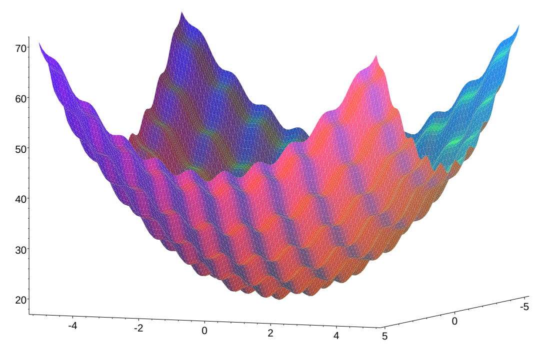


Figure 1: Rastrigin function for $n = 2$. The global optimum is located at $(x, y) = (0, 0)$.

The Rastrigin function has been minimized for $n = 20$, with a *Hybrid Genetic Algorithm*.

2.2 The Hybrid Genetic Algorithm

- Binary Coded GA :
  A standard GA has been used.

popsiz	50
maximal number of generations	150
p_c	0.85
p_m	0.003
binary tournament	
1-point crossover	
mutation at bit-level	
elitist strategy	

- The deterministic method :
  A steepest descent method has been selected, with a search of the optimal step at each gradient iteration by a 1D minimization.

  Initializing the steepest descent algorithm with the best element of the population obtained after 150 generations of the GA, eight gradient iterations were found sufficient to meet the convergence criterion.

2.3 Results

For each gradient iteration, the 1D minimization makes 3 calls to the analytical function evaluation. So, $50 + 50 \times 150 + 3 \times 8 = 7574$ function evaluations were made.

Figure 2 presents the convergence of the minmum of F as a function of the number of generations. The figure also provides a comparison between simple GA with the Hybrid GA. Of course, for this particular problem, the gradient-based method, here the steepest descent algorithm, would generally not converge to the global optimum unless the initial guess is made very close to it.

The optimal solution $(x_i)_{i=1\dots n}$ is exactly equal to 0.

3 A Multi-Objective problem

This test-case has been proposed by C. Poloni from the University of Trieste, in Italy [11]

3.1 The Multi-Objective problem

Let us maximize :

$$\begin{cases} F_1(x,y) = & -(1 + (A_1 - B_1)^2 + (A_2 - B_2)^2) \\ F_2(x,y) = & -(x+3)^2 - (y+1)^2 \qquad\qquad (x,y) \in [-\pi, \pi] \end{cases}$$

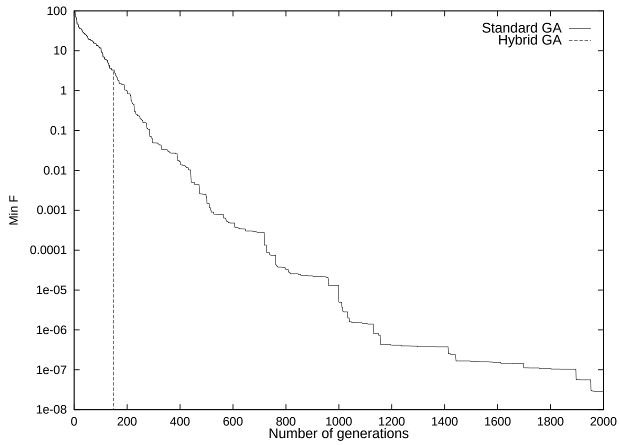


Figure 2: Convergence history of the Minimum of F, with a simple GA and with a Hybrid GA.

with $A_i = \sum_{j=1}^{2} a_{ij} \sin \alpha_j + b_{ij} \cos \alpha_j$ and $B_i = \sum_{j=1}^{2} a_{ij} \sin \beta_j + b_{ij} \cos \beta_j$

$$\text{and} \quad a = \begin{pmatrix} 0.5 & 1 \\ 1.5 & 2 \end{pmatrix} \quad b = \begin{pmatrix} -2 & -1.5 \\ -1 & -0.5 \end{pmatrix} \quad \alpha = (1 \;\; 2) \quad \beta = (x \;\; y).$$

This maximization problem can be solved with or without constraints. The constraints are of the following form :

$$(x-2)^2 + (y-2)^2 \leq 9$$
$$(x+2)^2 + (y+2)^2 \geq 9$$

When the number of parameters is greater than 2, x and y are generalized as follows :

$$x = \sum_{i=1}^{\frac{n}{2}} x_i \qquad y = \sum_{i=\frac{n}{2}+1}^{n} x_i \qquad x_i \in [-\frac{\pi}{\frac{n}{2}}, \frac{\pi}{\frac{n}{2}}] \qquad n = 2, 4, 8, 16$$

This multi-objective maximization problem is solved by a Non Dominated Sorting Genetic Algorithm (inspired from [13]).

3.2 Description of the optimizer

A general multi-objective optimization problem consists of a number of objectives to be optimized simultaneously and is associated with a number of inequality and/or equality constraints (for a theoretical background, see for example the book of Cohon [3]). It can be formulated as follows :

$$\text{Minimize / Maximize} \quad f_i(x) \quad i = 1, \cdots, N$$

$$\text{subject to :} \begin{cases} g_j(x) = 0 & j = 1, \cdots, M \\ h_k(x) \leq 0 & k = 1, \cdots, K \end{cases} \qquad \text{constraints}$$

The f_i are the objective functions, N is the number of objectives, x is a p-dimensional vector whose p components are known as decision variables. In a maximization problem, a vector x^1 is partially greater than another vector x^2 when :

$$\forall i \;\; f_i(x^1) \geq f_i(x^2) \quad (i = 1, \cdots, N) \text{ and there exists at least one } i \text{ such that} f_i(x^1) > f_i(x^2) \;.$$

We then say that *the solution* x^1 *dominates the solution* x^2.

Definition of non-dominance for 2 criteria in the case of a maximization :

$$\begin{cases} \text{Maximize} \quad f(x) = (f_1(x), f_2(x)) \\ \\ \text{such that} \quad x \in X, \text{ the feasible region.} \end{cases}$$

x^1 and x^2 are two solutions to be compared. Then, x^1 dominates x^2 iff :

$$f_1(x^1) > f_1(x^2) \quad \text{and} \quad f_2(x^1) \geq f_2(x^2) \quad \text{or} \quad f_1(x^1) \geq f_1(x^2) \quad \text{and} \quad f_2(x^1) > f_2(x^2)$$

The non-dominated individuals define the **Pareto front**.

Ranking by fronts

GAs operate primarily by selection of the individuals according to their fitness values. But, in a multi-objective optimization problem, several criteria are assigned and *not* a single fitness value. Therefore, to evaluate the individuals, we need to build a fitness value, called *dummy fitness.*

After the application of the definition of non-dominance, the chromosomes are classified by fronts. The non-dominated individuals define front 1 ; among the remaining individuals, the non-dominated ones define front 2, etc. The worst individuals define front f, where f is the number of fronts. Once the individuals have been ranked, they can be assigned the following dummy fitness values :

$$f_i = \textit{rank to which i belongs}$$

Let us note that the problem becomes, now, a **minimization problem !**

Our results in multi-objective optimization are based on the algorithm of Srinivas and Deb [13], called the Non-dominated Sorting Genetic Algorithm (NSGA). This method allows to obtain a representative sampling of solutions all along the Pareto front (see [8]).

Non Dominated Sorting Genetic Algorithms

The non-dominated sorting procedure requires a ranking selection method which emphasizes the optimal points. The sharing technique, or niche method, is used to stablilize the subpopulations of the "good" points.

We use this method, because one of the main defects of GAs in a multi-objective optimization problem is premature convergence. In some cases, GAs may converge very quickly to a point of the optimal Pareto set and as the associated solution is better than the others (it is called a "super individual"), it breeds in the population and, after a number of generations the population is composed of copies of this individual only ! The non-dominated sorting technique with the niche method avoids this situation because as soon as a solution is found in multiple copies, its fitness value is increased and in the next generation, new solutions that are different can be found, even if they are not so good.

Technical aspects of the method :

1. From an initial population of solutions, one first identifies the non-dominated individuals. These individuals belong to rank number 1 and they have large probabilities to be reproduced.

2. Then one assigns the same dummy fitness f_i to the non-dominated individuals i of front 1 (generally, the dummy fitness f_i is equal to 1).

3. To maintain the diversity, the dummy fitness value of the individuals is then increased (let us recall that the maximization problem has become a minimization because of the ranking by fronts) : it is multiplied by a quantity proportional to the number of individuals around it, according to a given radius. If the individual has a lot of neighbours, there is a lot of similar solutions. Then, the fitness is increased in order to create some diversity in the next generation.
   We call this quantity a **niche** and it is evaluated as follows :

   $$m_i = \sum_{j=1}^{M} Sh(d(i,j))$$

   where i is the number of the individual, M is the number of the individuals belonging to the current front (or rank).
   $Sh(d)$ is the **Sharing function**, a linear decreasing function of d defined by :

$$\begin{cases} Sh(0) = 1 \\ Sh(d(i,j)) = \begin{cases} 1 - \dfrac{d(i,j)}{\sigma_{share}} & \text{if } d(i,j) < \sigma_{share} \\ 0 & \text{if } d(i,j) \geq \sigma_{share} \end{cases} \end{cases}$$

σ_{share} is the maximum phenotypic distance allowed between any two individuals to become members of a niche. It has no universal value ; it is problem-dependent. $d(i,j)$ is the distance between two individuals i and j. It can define the *Hamming distance* (a genotypic distance, at string-level) or the *Euclidian distance* (a phenotypic distance, at real-level). Then after the niche has been created for each individual of the current front, a new dummy fitness $f_i * m_i$ is assigned.

4. After the above sharing process has been applied, the non-dominated individuals of Front 1 are temporarily ignored from the population. The following non-dominated individuals define Front 2. One evaluates the minimum of the quantities $f_i * m_i$ taken over all chromosomes of Front 1. Then this value is assigned as an initial dummy fitness to the individuals of Front 2 ; a niche is evaluated for each of them, and next, new dummy fitness values.

5. This process continues until the whole population has been visited.

6. As all the individuals in the population have now a dummy fitness value, selection, crossover and mutation can be applied. The flowchart of the method is presented in Figure 3.

3.3 Results

The number of individuals on the optimal Pareto fronts with and without constraints are presented in the following table :

Number of variables	2	4	8	16
Individuals on Pareto front without constraints	190	461	440	404
Individuals on Pareto front with constraints	505	442	590	248

In the case without constraints, the description of the Pareto front is realized by a larger number of individuals as the number of variables increases. However, this trend

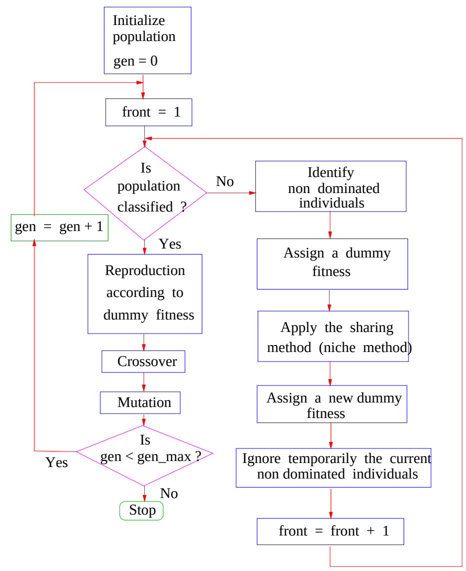


Figure 3: Flowchart of the Non Dominated Sorting GA.

is inverted in the case of constraints.

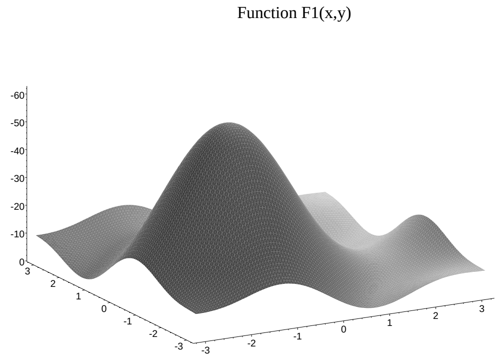
The parameters for the GA are :

popsiz	40
maximal number of generations	60
p_c	0.9
binary tournament	
1-point crossover	
mutation at bit-level	
elitist front	
euclidian distance	

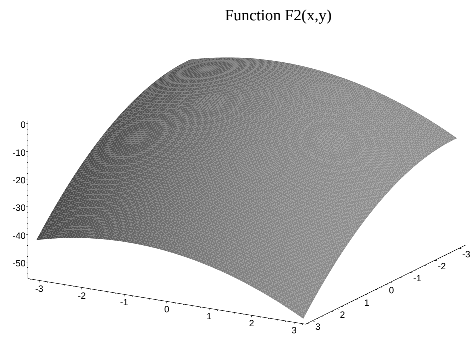
On Figures 4(a)-(b), the two analytical functions are depicted for $n = 2$, function of (x, y). Figure 5 represents the range of the application $(x, y) \mapsto (F_1, F_2)$.

3.3.1 Results for the case without constraints

In Figures 6(a)-(b)-(c)-(d), the optimal Pareto fronts are depicted. It took 4880 cost evaluations for each maximization.



(a) Function $F_1(x,y)$



(b) Function $F_2(x,y)$

Figure 4: Plot of the two analytical functions, function of (x,y), in the cartesian plane.

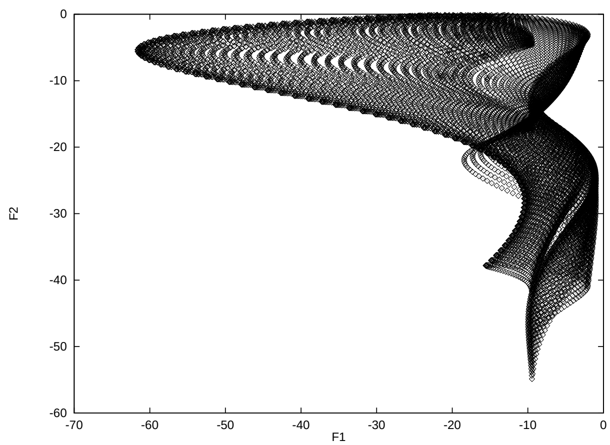
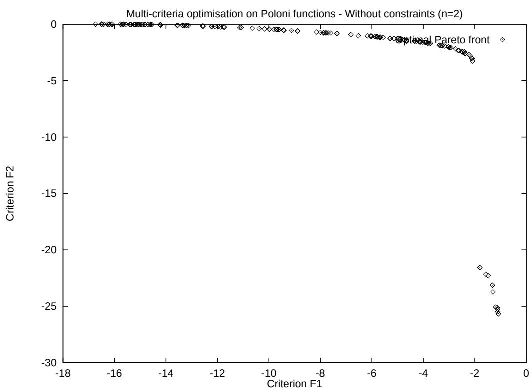


Figure 5: Representation of the range of the application :$(x,y) \mapsto (F_1,F_2)$.

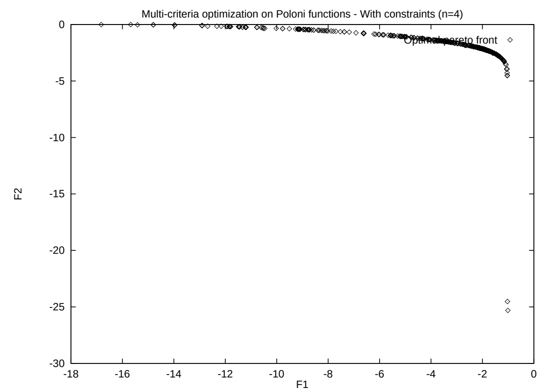
3.3.2 Results for the case with constraints

In Figures 7(a)-(b)-(c)-(d), the optimal Pareto fronts are depicted. It took 4880 cost evaluations for each maximization.

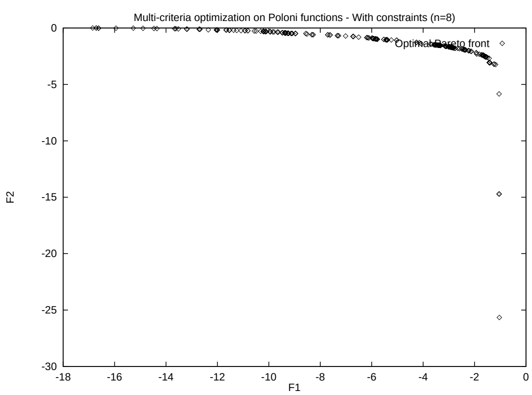
In both cases, the Pareto fronts are relatively well represented in the (F_1,F_2) plane, by virtue of the niching technique.



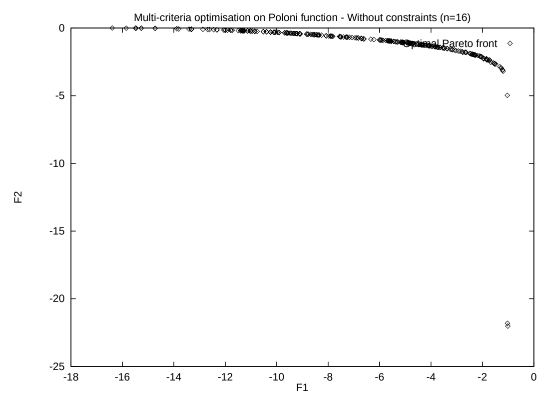
(a) $n = 2,\ p_m = 0.05$ and $\sigma_{share} = 4$



(b) $n = 4,\ p_m = 0.008$ and $\sigma_{share} = 4$

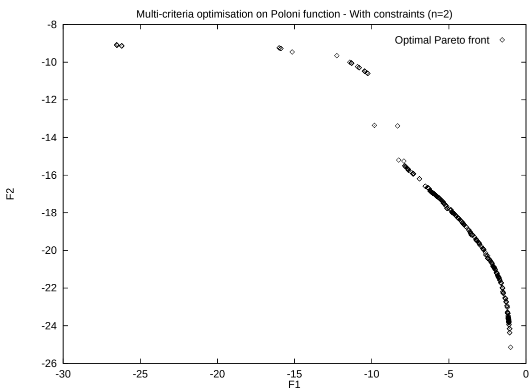


(c) $n = 8,\ p_m = 0.001$ and $\sigma_{share} = 2$

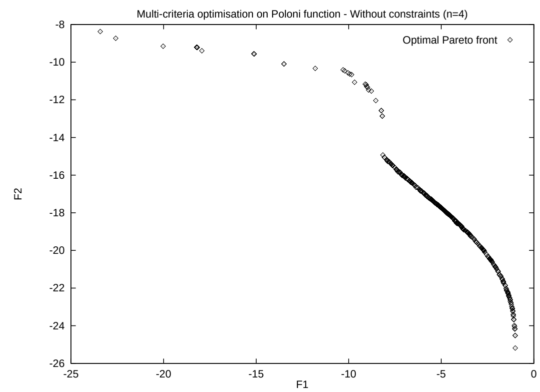


(d) $n = 16,\ p_m = 0.001$ and $\sigma_{share} = 2$

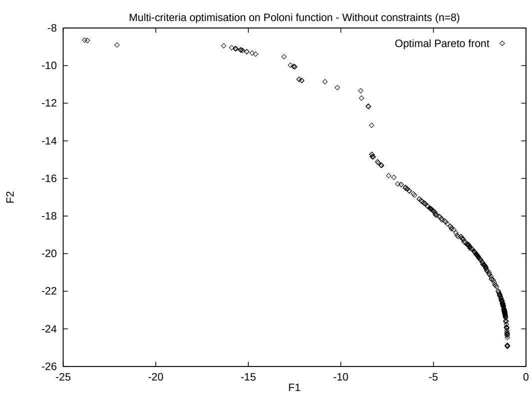
Figure 6: Optimal Pareto fronts when not using constraints.



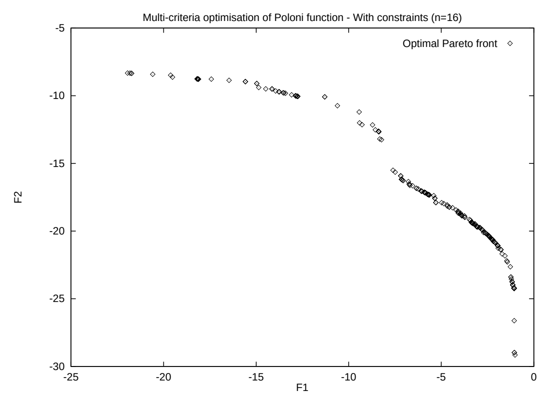
(a) $n = 2,\ p_m = 0.01$ and $\sigma_{share} = 2$



(b) $n = 4,\ p_m = 0.01$ and $\sigma_{share} = 2$



(c) $n = 8,\ p_m = 0.005$ and $\sigma_{share} = 2$



(d) $n = 16,\ p_m = 0.001$ and $\sigma_{share} = 2$

Figure 7: Optimal Pareto fronts when using constraints.

4 Keane's function

4.1 The maximization problem

Let us maximize the following function [6] :

$$F(x) = \left| \frac{\sum_{i=1}^{n} cos^4(x_i) - 2\prod_{i=1}^{n} cos^2(x_i)}{\sqrt{\sum_{i=1}^{n} i x_i^2}} \right|$$

with the two non-linear constraints :

$$\sum_{i=1}^{n} x_i \leq 7.5n \quad \text{and} \quad \prod_{i=1}^{n} x_i \geq 0.75$$

$(x_i)_{i=1\ldots n} \in [0, 10]$ and $n = 20, 50$.

Keane's function is depicted on Figure 8 for $n = 2$. It exhibits numerous local maxima.

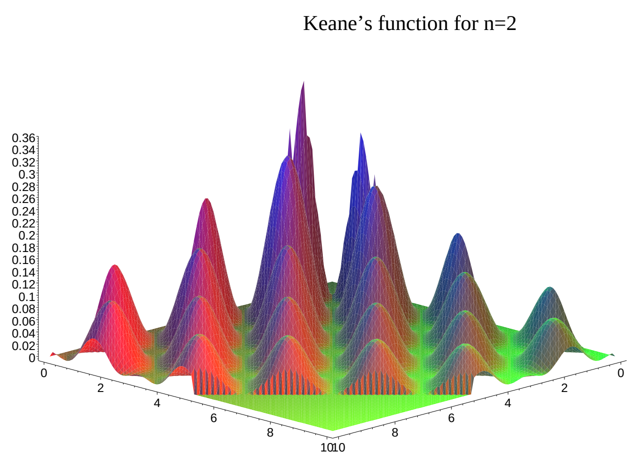


Figure 8: Keane's function for $n = 2$.

Note that the presence of an absolute value in the definition of the function makes it impossible to use a gradient-based optimizer.

4.2 The optimizer

A standard GA has been hybridized with a stochastic hill-climbing method (see [2]). The strategy is the following :

The best chromosome of the current generation is chosen as the starting point for the hill-climbing method. A parameter is chosen randomly and its value is perturbed by a small increment in either the plus or minus direction. A new chromosome is created. If it has a fitness value better than the previous one, the new chromosome replaces the previous one as the starting search point. If instead the fitness value of the new chromosome is not better, it is rejected. A new parameter is again chosen at random. The process continues until all the parameters have been perturbed and improvement is no longer achieved.

Our strategy was to alternate GA with hill-climbing.

4.3 Results

4.3.1 Case $n = 20$

The parameters for operating the GA were set as follows :

popsiz	80
maximal number of generations	1000
p_c	0.9
p_m	0.01
binary tournament	
1-point crossover	
mutation at bit-level	
elitist strategy	
niche method σ_{share}	0.34

Hill-climbing starts at the 350^{th} generation and it is applied on the best 10 individuals. The small perturbation is equal to $\varepsilon = 0.6$. We only succeeded to realize a moderate improvement of convergence to $F_{max} = 0.796$ whereas with the standard GA, the value 0.76 was achieved. Z. Michalewicz has converged to $F_{max} = 0.803553$ (see [9]) and T. Bäck converged to $F_{max} = 0.803619$ [1].
The optimal parameters are described in the following table.

x_1	3.07833	x_2	3.17595	x_3	3.05128
x_4	3.11311	x_5	3.00222	x_6	3.00749
x_7	2.90499	x_8	2.88128	x_9	0.631086
x_{10}	0.587335	x_{11}	0.517706	x_{12}	0.483315
x_{13}	0.410053	x_{14}	0.483996	x_{15}	0.497608
x_{16}	0.381690	x_{17}	0.497847	x_{18}	0.432111
x_{19}	0.427704	x_{20}	0.334261		

Figure 9 shows the convergence history of $-F$ as a function of generations.

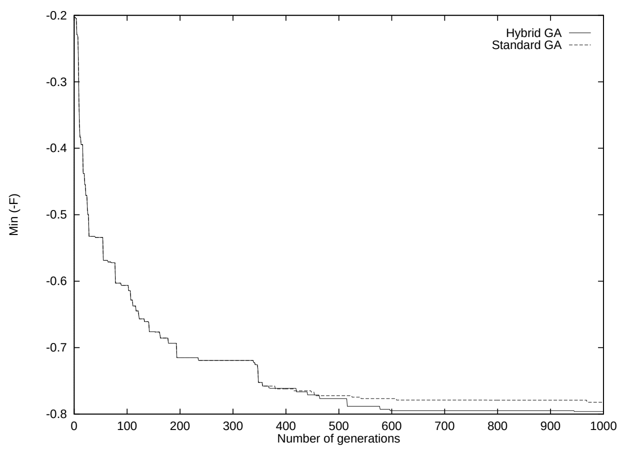


Figure 9: $\min(-F)$ for $n = 20$

4.3.2 Case $n = 50$

The parameters for operating the GA were set as follows :

popsiz	40
maximal number of generations	2000
p_c	0.8
p_m	0.006
binary tournament	
1-point crossover	
mutation at bit-level	
elitist strategy	
niche method σ_{share}	0.3

Hill-climbing starts at the 130^{th} generation and it is applied on the best 5 individuals. The perturbation is equal to $\varepsilon = 0.5$. Our very best result is $F_{max} = 0.796$ (with the standard GA, we converged to 0.777). Z. Michalewicz has converged to $F_{max} = 0.833$ (see [9]) while T. Bäck has converged to $F_{max} = 0.835$. The optimal parameters are described in the following table.

x_1	6.31922	x_2	3.13035	x_3	3.12450	x_4	3.25429
x_5	3.28268	x_6	3.26675	x_7	2.96492	x_8	2.99343
x_9	3.08870	x_{10}	3.01138	x_{11}	3.16188	x_{12}	2.99087
x_{13}	3.05707	x_{14}	2.92855	x_{15}	2.95711	x_{16}	3.14631
x_{17}	3.02403	x_{18}	3.13817	x_{19}	0.438648	x_{20}	2.92479
x_{21}	2.81879	x_{22}	2.90895	x_{23}	0.491447	x_{24}	0.694068
x_{25}	0.482017	x_{26}	0.314611	x_{27}	0.654735	x_{28}	0.311840
x_{29}	0.447506	x_{30}	0.558782	x_{31}	0.591541	x_{32}	0.510883
x_{33}	0.343749	x_{34}	0.486843	x_{35}	0.416858	x_{36}	0.632994
x_{37}	0.345275	x_{38}	0.449402	x_{39}	0.527740	x_{40}	0.634498
x_{41}	0.411452	x_{42}	0.340180	x_{43}	0.422506	x_{44}	0.275047
x_{45}	0.414413	x_{46}	0.433987	x_{47}	0.334957	x_{48}	0.566274
x_{49}	0.352310	x_{50}	0.321609				

Figure 10 depicts the convergence history of $-F$ as a function of the number of generations.

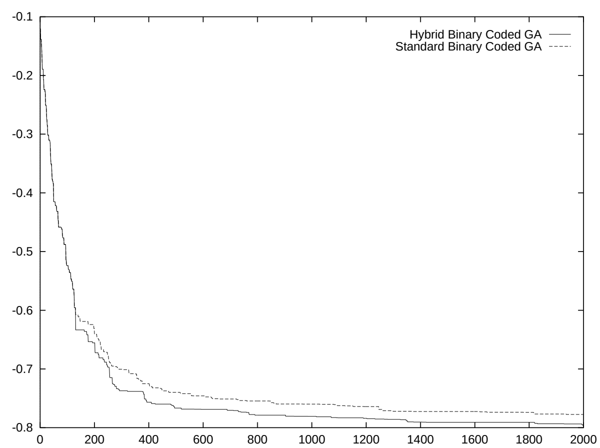


Figure 10: $\min(-F)$ for $n = 20$

5 Another Multi-Objective problem

This test-case is proposed by D. Quagliarella from CIRA, in Italy [12].

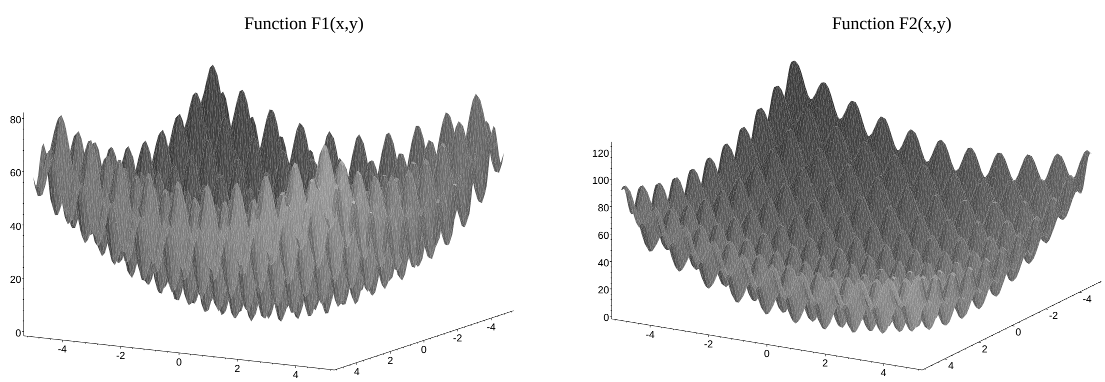
5.1 The Multi-Objective problem

Let us minimize $\quad F(x) = (F_1(x), F_2(x)) \quad$ with :

$$F_k(x) = \left(\frac{\sum_{i=1}^{n} ((x_i - a_k)^2 - 10\cos(2\pi(x_i - a_k)) + 10)}{n} \right)^{\frac{1}{4}} \qquad k = 1, 2$$

where $n = 30$, $x = (x_i)_{i=1\ldots n} \in [-5.12, 5.12]$, $a_1 = 0$ and $a_2 = 1.5$.

On Figures 11(a)-(b), F_1 and F_2 are plotted, function of (x, y).



(a) $F_1(x, y)$ (b) $F_2(x, y)$

Figure 11: Plots of F_1 and F_2 as a function of (x, y), for $n = 2$.

5.2 Description of the optimizer

The multi-objective minimization problem is also solved by the Non Dominated Sorting Genetic Algorithm used in section 3.

5.3 Results

The data of the GA are the following :

popsiz	60
maximal number of generations	250
p_c	0.9
p_m	0.008
σ_{share}	1.5
euclidian distance	
binary tournament	
1-point crossover	
mutation at bit-level	
elitist front	

On Figure 12 is depicted the evolution of the population during the generations, toward the optimal Pareto front.

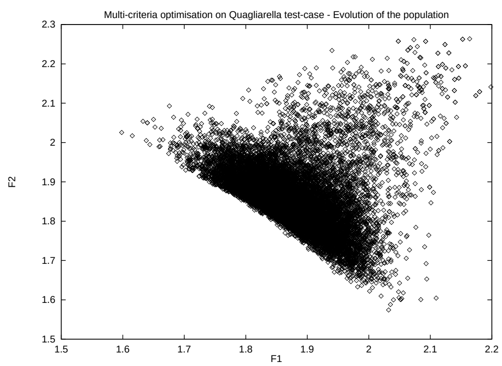


Figure 12: Evolution of the population : $2 \times (60 + (60 \times 250)) = 30120$ evaluations.

The optimal Pareto front is plotted on Figure 13.

6 Concluding remarks

GAs are very well suited for the optimization of very complex functionals (describing a lot of local optima).

Concerning the minimization of the Rastrigin function, our GA converges to the global optimum, but demands 100000 evaluations. When applying hybridization (GA

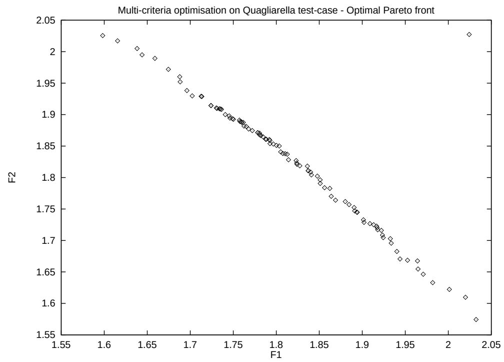


Figure 13: Optimal Pareto front with 96 individuals.

+ gradient), the global optimum is found very quickly, demanding only about 7600 evaluations. In the case where it is possible to evaluate the gradient of the functional, Hybrid GAs are more robust than standard GAs. GAs converge easily near the global optimum, but next, they are very slow ; then, a deterministic method is applied, more efficient near the optimum. If the functional is not differentiable, a hill-climbing method can be used as a deterministic one (see [2]).

In the case of Keane's function, a simple GA is not found robust enough to converge to the global optimum. We then have used a Hybrid GA (GA with hill-climbing), but we did not succeed to achieve the same level of convergence as Z. Michalewicz or T. Bäck (with an evolution strategy). In the case of $n = 20$ parameters, 210080 evaluations were necessary while for $n = 50$ parameters, 547540 evaluations were necessary ! Some improvement is then necessary... either for the GA (for example by improvement of the convergence by a "remainder stochastic sampling" from Goldberg[4] or a "steady state GA" from Syswerda[14],...) or by a more sophisticated hybridization process.

Concerning multi-objective optimization, the Non Dominated Sorting GA is well suited for obtaining a uniform Pareto front. It prevents from premature convergence. An improvement would be to find a value of σ_{share} in a more interactive way (for instance, it is tuned by hand). D. Goldberg proposes a solution in [5]. Note that optimizing/monitoring the parameters operating on a GA constitutes a vast subject of current research.

References

[1] T. Bäck, M. Laumanns, B. Naujoks, and L. Willmes. Test Case Computation Results, 1998. INGENET Project Report D5.1 (ICD).

[2] H.V. Cao and G.A. Blom. Navier-Stokes / Genetic Optimization of Multi-Element Airfoils. Technical Report 96-2487, AIAA Paper, New Orleans (LA), June 1996.

[3] J.L. Cohon. *Multiobjective Programming and Planning.* Academic Press, New-York, 1978. R. Bellman Editor.

[4] D.E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning.* Addison-Wesley Company INC., 1989.

[5] D.E. Goldberg and L. Wang. Adaptive niching via coevalutionary sharing. In D. Quagliarella, J. Périaux, C. Poloni, and G. Winter, editors, *Genetic Algorithms and Evolution Strategies in Engineering and Computer Science*, pages 21–38, Trieste (Italy), 1997. John Wiley and Sons.

[6] A. Keane. Genetic Algorithms digest, 1994. V8n16.

[7] N. Marco and S. Lanteri. A two-level parallelization strategy for Genetic Algorithms applied to shape optimum design, 1998. INRIA Report No 3463 - Submitted to *Parallel Comput.*

[8] N. Marco-Blaszka and S. Lanteri. Multi-Objective Optimization in CFD by Genetic Algorithms, 1999. INRIA Research Report in preparation.

[9] Z. Michalewicz, G. Nazhiyath, and M. Michalewicz. A note on usefulness of geometrical crossover for numerical optimization problems. In P.J. Angeline L.J. Fogel and T. Bäck, editors, *proceedings of the Fifth Annual Conference on Evolutionary Programming*, pages 305–312, Cambridge (MA), 1996. The MIT Press.

[10] H. Mühlenbein, M. Schomisch, and J. Born. The parallel genetic algorithm as function optimizer. *Parallel Computing*, 6-7(17):619–632, 1991.

[11] C. Poloni and V. Pediroda. GA coupled with Computationally Expensive Simulations : Tools to improve Efficiency. In D. Quagliarella, J. Périaux, C. Poloni, and G. Winter, editors, *Genetic Algorithms and Evolution Strategies in Engineering and Computer Science*, pages 267–288. John Wiley and Sons, 1997.

[12] D. Quagliarella and A. Vicini. INGENET Mathematical Test Problem T51.4, 1999. INGENET Technical Report.

[13] N. Srinivas and K. Deb. Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithm. *Evolutionary Computation*, 2(3):221–248, 1995.

[14] G. Syswerda. Uniform Crossover in Genetic Algorithms. In *Proceedings of the 3rd ICGA*, pages 1–9, Arlington (VA), 1989.

Unité de recherche INRIA Sophia Antipolis
2004, route des Lucioles - B.P. 93 - 06902 Sophia Antipolis Cedex (France)

Unité de recherche INRIA Lorraine : Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - B.P. 101 - 54602 Villers lès Nancy Cedex (France)
Unité de recherche INRIA Rennes : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)
Unité de recherche INRIA Rhône-Alpes : 655, avenue de l'Europe - 38330 Montbonnot St Martin (France)
Unité de recherche INRIA Rocquencourt : Domaine de Voluceau - Rocquencourt - B.P. 105 - 78153 Le Chesnay Cedex (France)

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, B.P. 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399